

CLAIMS:

We claim:

1. A connection pool management system comprising:
a connection pool configured to store a plurality of idle connections; and
a connection manager programmed for coupling to said connection pool and
further programmed to validate individual ones of said idle connections by issuing a
non-blocking input/output (I/O) operation to each of said individual ones of said idle
connections.
2. The system of claim 1, wherein said connection pool has an array configuration
wherein each element in said array configuration comprises a timestamp data member
and a reference to one of said idle connections.
3. The system of claim 1, wherein said connection pool comprises a configuration
for a last-in first-out (LIFO) ordering of said idle connections.
4. The system of claim 1, wherein said connection pool comprises a configuration
for storing a global timestamp indicating a time value when an oldest one of idle
connections had been added to said connection pool.

5. A connection pool management method comprising the steps of:
- responsive to adding a first one of a plurality of idle connections to a connection pool, recording a global timestamp to indicate a time value when said first idle connection had been added to said connection pool;
 - responsive to adding subsequent ones of said idle connections to said connection pool, recording individual timestamps in said connection pool in association with corresponding ones of said subsequent ones of said idle connections;
 - comparing a timestamp of an oldest one of said idle connections to said global timestamp to determine whether a timeout condition has arisen; and,
 - responsive to determining that a timeout condition has arisen, probing at least one of said idle connections with a non-blocking input/output (I/O) request in order to validate said at least one of said idle connections.
6. The method of claim 5, wherein said comparing step comprises the step of comparing a timestamp associated with a least recently added connection in said connection pool to said global timestamp to determine whether a timeout condition has arisen.
7. The method of claim 5, wherein said comparing step comprises the steps of:
- summing said global timestamp and a pre-established timeout value;
 - comparing said timestamp of said oldest one of said idle connections with said summed global timestamp and pre-established time out value; and,

if said timestamp exceeds of said oldest one of said idle connections exceeds said summed global timestamp and pre-established timeout value, concluding that a timeout condition has arisen.

8. The method of claim 5, wherein said probing step comprises the steps of:
sequentially performing timestamp comparisons with said global timestamp for each of said idle connections in said connection pool; and,
for each one of said idle connections having a timestamp which exceeds a combination of said global timestamp and a pre-established timeout value, attempting a non-blocking I/O operation over said one of said idle connections.

9. The method of claim 5, further comprising the step of updating said global timestamp with a new value subsequent to probing said at least one of said idle connections.

10. The method of claim 5, further comprising the step of provisioning a most recently used one of said idle connections responsive to a request to provision an idle connection in said connection pool.

11. The method of claim 10, wherein said provisioning step further comprises the steps of:
probing said provisioned idle connection with a non-blocking input/output (I/O) request in order to validate said provisioned idle connection; and,

where said provisioned idle connection fails to validate, removing said provisioned idle connection from said connection pool and provisioning another one of said idle connections.

12. The method of claim 5, further comprising the step of adding each of said idle connections to said connection pool in a last-in first-out (LIFO) manner.

13. The method of claim 12, wherein said adding step further comprises the step of recording a timestamp in said connection pool for each added one of said idle connections to indicate when said added one of said idle connections had been added to said connection pool.

14. The method of claim 5, further comprising the step of removing from said connection pool each idle connection which fails validation in said probing step.

15. A machine readable storage having stored thereon a computer program for connection pool management, the computer program comprising a routine set of instructions which when executed by the machine cause the machine to perform the steps of:

responsive to adding a first one of a plurality of idle connections to a connection pool, recording a global timestamp to indicate a time value when said first idle connection had been added to said connection pool;

responsive to adding subsequent ones of said idle connections to said connection pool, recording individual timestamps in said connection pool in association with corresponding ones of said subsequent ones of said idle connections;

comparing a timestamp of an oldest one of said idle connections to said global timestamp to determine whether a timeout condition has arisen; and,

responsive to determining that a timeout condition has arisen, probing at least one of said idle connections with a non-blocking input/output (I/O) request in order to validate said at least one of said idle connections.

16. The machine readable storage of claim 15, wherein said comparing step comprises the step of comparing a timestamp associated with a least recently added connection in said connection pool to said global timestamp to determine whether a timeout condition has arisen.

17. The machine readable storage of claim 15, wherein said comparing step comprises the steps of:

summing said global timestamp and a pre-established timeout value;

comparing said timestamp of said oldest one of said idle connections with said summed global timestamp and pre-established time out value; and,

if said timestamp exceeds of said oldest one of said idle connections exceeds said summed global timestamp and pre-established timeout value, concluding that a timeout condition has arisen.

18. The machine readable storage of claim 15, wherein said probing step comprises the steps of:

sequentially performing timestamp comparisons with said global timestamp for each of said idle connections in said connection pool; and,

for each one of said idle connections having a timestamp which exceeds a combination of said global timestamp and a pre-established timeout value, attempting a non-blocking I/O operation over said one of said idle connections.

19. The machine readable storage of claim 15, further comprising the step of updating said global timestamp with a new value subsequent to probing said at least one of said idle connections.

20. The machine readable storage of claim 15, further comprising the step of provisioning a most recently used one of said idle connections responsive to a request to provision an idle connection in said connection pool.

21. The machine readable storage of claim 20, wherein said provisioning step further comprises the steps of:

probing said provisioned idle connection with a non-blocking input/output (I/O) request in order to validate said provisioned idle connection; and,

where said provisioned idle connection fails to validate, removing said provisioned idle connection from said connection pool and provisioning another one of said idle connections.

22. The machine readable storage of claim 15, further comprising the step of adding each of said idle connections to said connection pool in a last-in first-out (LIFO) manner.

23. The machine readable storage of claim 22, wherein said adding step further comprises the step of recording a timestamp in said connection pool for each added one of said idle connections to indicate when said added one of said idle connections had been added to said connection pool.

24. The machine readable storage of claim 15, further comprising the step of removing from said connection pool each idle connection which fails validation in said probing step.